Thompson, Dale R. (1998). "Comparison of Two Swap Heuristics with a Genetic Algorithm for the Design of an ATM Network," Kia Makki, Imrich Chlamtac, and Niki Pissinou, Editors, *7th International Conference on Computer Communications and Networks, October 12-15, 1998, Lafayette, Louisiana,* The Institute of Electrical and Electronics Engineers, Inc., Los Alamitos, CA, 833-837.

Comparison of Two Swap Heuristics with a Genetic Algorithm for the Design of an ATM Network

Dale R. Thompson
USAE Waterways Experiment Station
CEWES-IM-C
3909 Halls Ferry Road
Vicksburg, MS 39180

Griff L. Bilbro
North Carolina State University
PO Box 7911
Raleigh, NC 27695

Abstract

The challenge of a network topology design is to provide a configuration with minimum cost given specified constraints. Network topology design is NP-hard [1] and known algorithms to solve these problems run in time that increases exponentially with the number of choices. The economic importance of determining the placement of switches in an ATM network justifies heuristic methods to find a good configuration within a reasonable amount of time. In this paper, two types of heuristic algorithms are compared. The first algorithm is based on swapping used switch locations with unused switch locations. The second algorithm is a genetic algorithm.

1 INTRODUCTION [1]

The challenge of network design is to provide a good configuration that is cost effective and provides high performance for information services [2]. The optimization of a network belongs to a class of problems called NP-hard, so that no known algorithms run in polynomial time. The known algorithms to solve these problems run in exponential time with respect to the number of choices [1]. In a network design the costs are not linear because the capacities of the links and switches must be selected from a finite discrete set. Such hard combinatorial problems are usually solved using heuristics [3].

[1] The tests described and the resulting data presented herein, unless otherwise noted, were obtained from research conducted under the Laboratory Discretionary Research and Development Program of the United States Army Corps of Engineers by the Waterways Experiment Station. Permission was granted by the Chief of Engineers to publish this information.

In the topology, capacity, and flow assignment problem (TCFA) for networks, the traffic demand between geographic points is given and the problem is to define the topology of the network, to specify the capacities of the links and switches, and to determine the routes of the traffic flows of information that minimize the cost of the network under given constraints. Often the constraint is formulated as maximum average delay. Specific traffic flows result from the routing algorithm used, such as the shortest path algorithm which routes a flow over one path, or optimal routing that routes a flow over multiple paths. Because a cost function for TCFA typically exhibits several local minima, exact solutions can only be calculated for unrealistically small networks or unrealistically large run times. Heuristics are required to approximate the global solution for any but the smallest networks [4]. These heuristics are initialized at some feasible topology; then they repeatedly replace the current solution with some modification that improves the cost and preserves feasibility.

Such heuristics can be applied to problems with diverse routing algorithms, constraint inequalities, and cost functions, but the heuristic itself is characterized by the set of trial networks that are to be considered given the current topology and how the trial network is selected. A trial topology might be generated by identifying under-utilized links for elimination or capacity reduction. Or the capacities of over-utilized links could be increased to meet the delay constraints in some trial topology [3]. The greedy-drop heuristic successively deletes the least utilized node to find the minimum number of nodes that will support the traffic [5].

A node or branch can be exchanged to modify the topology of the network. The branch exchange method deletes a link between two nodes and adds a link between two different nodes [6]. The cut-saturation method identifies candidate branches by finding the minimum number of highly utilized branches that would divide the network into two separate networks if deleted. The capacity of these branches is increased or links are added between the two groups of nodes to modify the topology of the network. A node exchange deletes a node and adds a node. These methods can be repeated with a new starting topology or a different order of the same exchanges to look for a trial network with lower cost [7].

Several heuristic algorithms have been reported for designing asynchronous transfer mode (ATM) networks. Lo [8] develops a column-generation-based heuristic that uses node dimensioning and allows multipath routing for traffic in the same class. Numerous local minima occur when optimizing capacities and flows of virtual paths (VP) in an ATM network. Gerla, Monteiro, and Pazos [9] formulate the capacity and flow assignment problem for the VP topology problem and use the Frank-Wolfe steepest descent method to find a local minimum from several random initial solutions; the solution with the lowest average delay is chosen. Mitra, Morrison, and Ramakrishnan [10] formulate the optimization of an ATM network at the call-level as a multirate, circuit-switched, loss network with equivalent capacity requirements. They calculate the loss probabilities using the fixed-point approach assuming link independence. This approach terminates at a local minimum, but they report satisfactory results by repeating the procedure for several random initializations.

The key to approximating a solution is efficiently evaluating a given topology and generating good initial topologies [4]. One way to construct a feasible topology is to assign the maximum allowable capacity to each link and node [8]. Kershenbaum, Kermani, and Grover present a heuristic for generating low-cost trees that they suggest as a starting topology [11]. Dirilten and Donaldson generate feasible low-cost trees using linear regression clustering which could be used to initialize a network design heuristic [12]. A clustering algorithm called NEWCUST determines candidate concentrator locations by creating a list of K nearest neighbors for each node and determining which sites show up in the lists most frequently [13]. A review of methods to generate network topologies that have properties of real networks is presented in [14].

The genetic algorithm (GA) is another optimization technique for approximating the global solution to a complex problem space. It is based on "natural selection" of competing solutions and "genetic" encoding of each of those solutions. A population of individuals, each defined by a set of chromosomes that represent a solution, is ranked according to the fitness of the individual. Individuals are selected for reproduction, crossover, and mutation based on their relative fitness. A "biased" roulette wheel based on the fitness of the individual determines which individuals occur most frequently in the next generation. This is repeated for several generations so that each generation tends to produce fitter solutions. GA was used to expand existing computer networks while optimizing reliability in [15]. A GA for minimizing the average network delay of a spanning tree bridge network was presented in [16]. The topology of a wide-area network was optimized with GA in [17]. The embedded ATM topology on digital cross-connects was optimized with GA in [18].

The rest of this paper is organized as follows: In Section 2, we state the problem of optimally locating switches in ATM networks. Then, we develop a design heuristic algorithm based on swapping used ATM switch locations with unused ATM switch locations in Section 3. In Section 4, we present a genetic algorithm for optimizing the locations of ATM switches. We compare the two versions of swap algorithms with genetic algorithm for a prototype problem with fiber, traffic, and switches in Section 5.

2 STATEMENT OF THE PROBLEM

The problem is to minimize the cost of a network given certain constraints on quality of service (QoS) parameters for different classes of traffic flows. The algorithm must input and consider the geographic locations of traffic sources, the expected traffic flows between these sources, the existing fiber plant, and the cost and capacity of available switches. The proposed algorithm will route the traffic, specify the location of the switches, select the types of ATM switches from a given list, specify the number of fibers required from the existing fiber plant for this network, and specify the capacity for each link to minimize the total cost, which is also computed.

We make the following assumptions. Traffic sources will be connected to the ATM switch that has the least expensive path cost. Switch-to-switch traffic will be routed with a shortest path algorithm. ATM switches can be located only at a subset of given patch panel locations for the fiber optic physical plant. The ATM network will serve a campus environment under a single administrator so that the total cost of ATM switches is relevant and therefore ATM switches will not necessarily be placed at every node. The capacity of the links will be dimensioned in terms of effective bandwidth as follows: the effective bandwidth of the switch-to-switch links will be calculated by finding the minimum of the flow model and Gaussian model. The effective bandwidth of the source-to-switch traffic will be calculated with the flow model [19].

Each ATM switch may have multiple capacity ports. Traffic will be assigned to a switch by sorting the effective bandwidth of the links in decreasing order and assigning them to the ports in decreasing order of capacity. The capacity of all links will be symmetric.

Once the locations of switches are determined, the largest capacity switch will be assigned at each node. Then, a greedy heuristic algorithm will be applied at each node to determine the least cost switch that can satisfy the QoS constraints.

Given the assumptions above the problem becomes deciding which nodes will have switches. If there are N number of nodes and S switches, then each node could be in one of $S+1$ states. It could either have no switch or have one of the S types of switches. Therefore, there are $(S+1)^N - 1$ combinations that have at least one switch. For a network with fifteen nodes and two types of switches this is $14,348,906$ possible combinations that have at least one switch!

3 SWAP ALGORITHM

We developed and implemented two swap algorithms. The first is named SWAP1 w/K and the second is SWAP2 w/K. Both swap algorithms use a clustering algorithm to determine the candidate locations for ATM switches. This algorithm creates lists of K nearest neighbors for each node and determines the locations that appear in the lists most frequently [13]. Then, nodes are assigned to the candidate switch location that has the least expensive path cost. Finally, the two swap algorithms attempt to lower the cost of the network by swapping used and unused switch locations in the candidate list. The only difference between SWAP1 w/K and SWAP2 w/K is that SWAP2 w/K attempts to delete nodes before doing the random swaps. It successively deletes the highest cost per bandwidth node until the minimum number of nodes that will support the traffic with the given switch locations is found.

3.1 SWAP1 w/K

1. Find candidate switch sites with a clustering algorithm that as a parameter finds K nearest neighbors for each node [13].
2. Assign traffic to the switch that has the least expensive path cost, assign the largest switch to all used sites, and find routes using a shortest path algorithm.
3. Assign switches at each used node in a greedy fashion until the cheapest switch that supports the traffic is found.
4. Evaluate the network topology cost. If cheaper than the best, save it.
5. Swap a used and unused switch site randomly.
6. Go to step 2 until the maximum number of iterations is reached.
7. Output the cheapest topology.

4 GENETIC ALGORITHM

An algorithm which we named GENETIC was modified from [20] and implemented. In the chromosome a "1" represents a candidate switch location and a "−1" represents a node without a switch. Also, since the code was designed for maximizing functions with positive values the fitness value was set to $MAX\_COST - cost$ to minimize the cost.

1. Initialize the population with $POPSIZE$ random members.
2. Evaluate the fitness of the population.
3. Use the elitist strategy.
4. Select members for the next generation by using standard proportional selection. Find the cumulative fitness of each member and spin a "biased" roulette wheel.
5. Crossover two members with probability $PXOVER$.
6. Mutate the gene of a member with probability $PMUTATION$.
7. Evaluate the fitness of the population.
8. Use the elitist strategy.
9. Go to step 4 for the maximum number of generations, $MAXGENS$.
10. Output the cheapest topology.

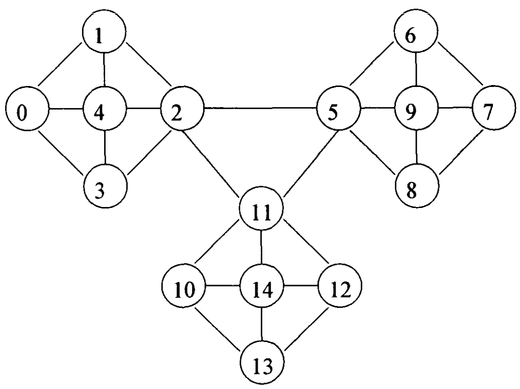

Figure 1. Fiber topology

O D	PCR	SCR	maxCTD	CLR
0,3	350K	350K	30 ms	10^{-5}
0,4	350K	350K	30 ms	10^{-5}
0,5	350K	350K	30 ms	10^{-5}
0,7	350K	350K	30 ms	10^{-5}
4,14	350K	350K	30 ms	10^{-5}

Table 1. Offered Load

5 PERFORMANCE EVALUATION

A prototype set of fiber, traffic, and available switches was developed to test the network topology algorithms. The fiber topology connects fifteen nodes and is shown in Figure 1. The offered traffic is shown in Table 1. It is assumed that there are two types of switches available. One switch costs $35,000 and has four 622 Megabit per second (Mbps) ports. The second switch costs $3,500 and has one 622 Mbps port and four 155 Mbps ports.

The SWAP1 w/K, SWAP2 w/K, and GENETIC algorithms were evaluated using the prototype set of fiber, traffic, and switches. The parameters were adjusted so that the number of times that common calculations were done was equivalent in all algorithms. All algorithms performed these steps 200 times. Each algorithm was run with the same 10 seeds for the random number generator. In the SWAP algorithms the seed determined the nodes and the order of nodes to be swapped. In the GENETIC algorithm the seed determined the members of the initial population. The SWAP1 w/K and SWAP2 w/K algorithms were run with K equal to 5 and 7. The GENETIC algorithm was run with $POPSIZE = 20$, $MAXGENS = 10$, $PXOVER = 0.8$, and $PMUTATION = 0.15$.

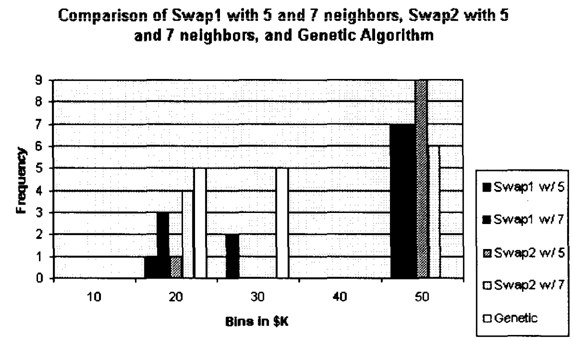

Figure 2. Histogram of Network Topology Costs

6 CONCLUSIONS

The lowest cost network topology discovered is $\$15.1K$ when ATM switches are placed at nodes 3, 4, and 14. A histogram of the cost of the network topologies is shown in Figure 2. The bins are in increments of $\$10K$. The best SWAP algorithm for this particular set of values was the SWAP2 w/7 algorithm which discovered a network topology near $\$15.1K$ four out of ten times, but the other six out of ten times the topology was near $\$41K$. Note that all the best topologies resulting from the GENETIC algorithm are in the bins $\$10K - \$20K$ or $\$20K - \$30K$. Therefore, it gave more consistent results for the different random number generator seeds than the SWAP algorithms. This demonstrates that the GENETIC algorithm has the potential to discover a lower cost network than the SWAP1 w/K and SWAP2 w/K algorithms. This is due to the GENETIC algorithm's inherent step of generating several initial random topologies when initializing the members of the population.

References

[1] B. Golden, M. Ball, and L. Bodin, "Current and future research directions in network optimization," *Computers & Operations Research*, vol. 8, no. 2, 1981, pp. 71-81.

[2] K.Y. Jo, M.P. Scher, and J.R. Mitchell, "Modeling and analysis of the defense information infrastructure," *Information and Decision Technologies*, vol. 19, 1994, pp. 577-594.

[3] D. Bertsekas and R. Gallager, *Data Networks 2nd ed.*, pp. 363-492, Upper Saddle River, NJ: Prentice Hall, 1992.

[4] L. Kleinrock, *Queueing Systems Vol. II: Computer Applications*, pp. 314-360, New York, NY: John Wiley & Sons, 1976.

[5] B. Gavish, "Topological design of computer communication networks-the overall design problem," *European Journal of Operational Research*, vol. 58, 1992, pp. 149-172.

[6] M. Gerla and L. Kleinrock, "On the topological design of distributed computer networks," *IEEE Transactions on Communications*, vol. 25, 1977, pp. 48-60.

[7] R.R. Boorstyn and H. Frank, "Large-scale network topological optimization," *IEEE Transactions on Communications*, vol. 25, 1977, pp. 29-47.

[8] S. Z. Lo, "ATM network topological design and network modification," PhD thesis, Operations Research, North Carolina State University, 1996.

[9] M. Gerla, J.A.S. Monteiro, and R. Pazos, "Topology design and bandwidth allocation in ATM nets," *IEEE Journal on Selected Areas in Communications*, vol. 7, no. 8, Oct. 1989, pp. 1253-1262.

[10] D. Mitra, J.A. Morrison, and K.G. Ramakrishnan, "ATM network design and optimization: a multirate loss network framework," *IEEE/ACM Transactions on Networking*, vol. 4, no. 4, Aug. 1996, pp. 531-543.

[11] A. Kershenbaum, P. Kermani, and G.A. Grover, "MENTOR: an algorithm for mesh network topological optimization and routing," *IEEE Transactions on Communications*, vol. 39, no. 4, April 1991, pp. 503-513.

[12] H. Dirilten and R.W. Donaldson, "Topological design of distributed data communication networks using linear regression clustering," *IEEE Transactions on Communications*, vol. 25, 1977, pp. 1083-1092.

[13] H.G. Dysart and N.D. Georganas, "Newclust: an algorithm for the topological design of two-level, multidrop teleprocessing networks," *IEEE Transactions on Communications*, vol. 26, no. 1, Jan. 1978, pp. 55-62.

[14] E. Zegura, K. Calvert, and M. Donahoo, "A quantitative comparison of graph-based models for internet topology," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, Dec. 1997, pp. 770-783.

[15] A. Kumar, R.M. Pathak, and Y.P. Gupta, "Genetic-algorithm-based reliability optimization for computer network expansion," *IEEE Transactions on Reliability*, vol. 44, no. 1, Mar. 1995, pp. 63-72.

[16] R. Elbaum and M. Sidi, "Topological design of local-area networks using genetic algorithms," *IEEE/ACM Transactions on Networking*, vol. 4, no. 5, Oct. 1996, pp. 766-778.

[17] S. Pierre and G. Legault, "A Genetic Algorithm for Designing Distributed Computer Network Topologies," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 28, no. 2, Apr. 1998, pp. 249-258.

[18] K. Tang, K. To, K. Man, and S. Kwong, "Topology design and bandwidth allocation of embedded ATM networks using genetic algorithm," *IEEE Communications Letters*, vol. 2, no. 6, Jun. 1998, pp. 171-173.

[19] R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, 1991, pp. 968-981.

[20] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs - 3rd rev. and extended ed.*, New York, NY: Springer-Verlag, 1996.